

Contents

- 1 Bonding
- 2 Samples
 - ◆ 2.1 Wifi
(Atheros)
 - ◇ 2.1.1
1st
device
(AP)
 - ◇ 2.1.2
2nd
device
(Sta)
 - ◆ 2.2
theoretical
consideration
- 3 Bonding Types
 - ◆ 3.1
balance-rr
 - ◆ 3.2
active-backup
 - ◆ 3.3
balance-xor
 - ◆ 3.4
broadcast
 - ◆ 3.5 802.3ad
 - ◆ 3.6
balance-tlb
 - ◆ 3.7
balance-alb
 - ◆ 3.8
weighted-rr
 - ◆ 3.9 duplex
- 4 External links

Bonding

location :

<http://ip/Networking.asp>

Samples

This is a description to setup a sample p2p-bridge with 2 bonded interfaces.

Bonding

You can use this guide to see how to do the right settings.

Wifi (Atheros)

Be aware that you cant encrypt this connection! This will not work.

The chance to get a encrypted link might be to use 2 EoIP/VLAN's accross the links and bond them.

1st device (AP)

- Set your wlan interfaces to WDS-AP (see [WDS#Atheros](#)).
- Under "Create Bridge" create 2 bridge devices and call them e.g. br1 and br2 (dont use a existing name!), but dont set IP's!
- Under "Assign to Bridge" add each of your 2 wlan interfaces (athX) to one of the created bridges from above
- Choose the bonding typ u wanna use (read below).
- Create a bonding with the 2 bridge interfaces from above and assign them to the setup
- Assign the bond (bond0) to the default existing brigde (br0)

2nd device (Sta)

- Set your wlan interfaces to WDS-Sta (see [WDS#Atheros](#)).
- Unbridge every interface device, but dont set IP's!
- Choose the bonding typ u wanna use (read below).
- Create a bonding with the 2 interfaces and assign them to the setup
- At least assign the bond (bond0)to the brigde (br0)

theoretical consideration

bonding 2 wifi links with encryption should also work if you will use it with eoip over wifi like this:

- Set your wlan interfaces to WDS (see [WDS#Atheros](#)), set encryption, unbridge both sides and set 4 different ip's on the 4 interfaces.
- create 2 unbridged eoip tunnels which connect the 2 links, but dont set IP's!
- now follow setup like above to setup the bonding on the 2 eoip tunnels

Bonding Types

balance-rr

Round-robin policy: Transmit packets in sequential order from the first available slave through the last. This mode provides load balancing and fault tolerance.

active-backup

Active-backup policy: Only one slave in the bond is active. A different slave becomes active if, and only if, the active slave fails. The bond's MAC address is externally visible on only one port (network adapter) to avoid confusing the switch. This mode provides fault tolerance. The primary option affects the behavior of this mode.

balance-xor

XOR policy: Transmit based on [(source MAC address XOR'd with destination MAC address) modulo slave count]. This selects the same slave for each destination MAC address. This mode provides load balancing and fault tolerance.

broadcast

Broadcast policy: transmits everything on all slave interfaces. This mode provides fault tolerance.

802.3ad

IEEE 802.3ad Dynamic link aggregation aka LACP. Creates aggregation groups that share the same speed and duplex settings. Utilizes all slaves in the active aggregator according to the 802.3ad specification.

Pre-requisites:

1. Ethtool support in the base drivers for retrieving the speed and duplex of each slave.
2. A switch that supports IEEE 802.3ad Dynamic link aggregation. Most switches will require some type of configuration to enable 802.3ad mode.

balance-tlb

Adaptive transmit load balancing: channel bonding that does not require any special switch support. The outgoing traffic is distributed according to the current load (computed relative to the speed) on each slave. Incoming traffic is received by the current slave. If the receiving slave fails, another slave takes over the MAC address of the failed receiving slave.

Prerequisite: Ethtool support in the base drivers for retrieving the speed of each slave.

balance-alb

Adaptive load balancing: includes balance-tlb plus receive load balancing (rlb) for IPV4 traffic, and does not require any special switch support. The receive load balancing is achieved by ARP negotiation. The bonding driver intercepts the ARP Replies sent by the local system on their way out and overwrites the source

Bonding

hardware address with the unique hardware address of one of the slaves in the bond such that different peers use different hardware addresses for the server.

weighted-rr

Weighted round-robin bonding. In this mode bonding interface will use weights assigned to it's slaves. Each slave can have weight assigned via ioctl (ifenslave). These values will be used at the start of each "cycle". Each slave will have token counter restored to it's weight. Then using round-robin mechanism those tokens are "used" to pay for emitted frames. When all token counters are zeroed - new "cycle" begins.

duplex

Uses one dedicated link for TX and one for RX. So u will have a Full-Duplex connection

External links

<http://sourceforge.net/projects/bonding/files/>

<http://www.linuxhorizon.ro/bonding.html>